Reduced order adaptive models for systems of PDEs using POD

F. Terragni, M.L. Rapún and J.M. Vega

Fundamentos Matemáticos, Universidad Politécnica de Madrid, Spain

Sevilla, April 2011

Main goal: to accelerate numerical codes

Current trend to replace wind tunnel tests by CFD in Industrial Aerodynamics, but CFD of 3D problems at large Re may require non affordable resources. Same trend in other fields

Example (P.Moin'04):

- Commercial aircraft, 50 meter-long fuselage, cruising at 250m/s at an altitude of 10000m. Re$\approx 10^7$
- DNS requires 10^{16} grid points and $\Delta t = 10^{-4}$ seconds
- Current algorithms and software with a supercomputer (10^{12} floating–point operations per second) would take several thousand years to compute the flow for just one second of flight time!

Main goal: to accelerate numerical codes

Cheaper (but rough) alternatives

- RANS approximates cruise condition as steady states (2CPU days in a PC)
- Panel method (1 CPU minute)
- Aerodynamics Vortex Lattice methods (1 CPU second)

Are there better (cheaper, precise) alternatives?

- Better algorithms
- better software/hardware
- reduced order models

Main goal: to accelerate numerical codes

- Additional difficulty: design and certification (as other industrial/scientific problems do) involve many parameters that exponentially enlarge computational effort.
- Good news: The required precision is usually not large in industrial problems
- Industrial solvers are usually rough (coarse meshes, unphysical terms/BCs added to accelerate or stabilize), but stil require a large number of mesh points/modes/time steps
- Numerical complexity (degrees of freedom) is larger than the physical complexity of the flow (qualitatively different spatio–temporal features): a description in terms of a few modes should be possible

Outline

1. POD + Galerkin projection
   - POD + Galerkin projection (attractors)
   - POD + local Galerkin projection (transients+attractors)
2. Application to two model problems
   - The complex Ginzburg–Landau equation
   - The pulsating cavity
3. Conclusions and current work
   - Improving the method
   - Conclusions

Outline

1. POD + Galerkin projection
   - POD + Galerkin projection (attractors)
   - POD + local Galerkin projection (transients+attractors)
2. Application to two model problems
   - The complex Ginzburg–Landau equation
   - The pulsating cavity
3. Conclusions and current work
   - Improving the method
   - Conclusions

POD + Galerkin projection

Proper orthogonal decomposition (POD)

- POD of vectors provides an optimal basis (RMS) of the vector span
- If redundancies (due to, e.g. physical laws), the dimension of the truncated POD basis is much smaller

Main idea in two steps

1. Numerical calculation of some representative snapshots (Sirovich '78) in a time interval
2. Galerkin projection of the system of PDEs on the most energetic POD modes.

POD + Galerkin projection

Advantages

- Reduces the dimension of the problem
- Can be used for parameter values not considered in the snapshot set calculation

Dificulties

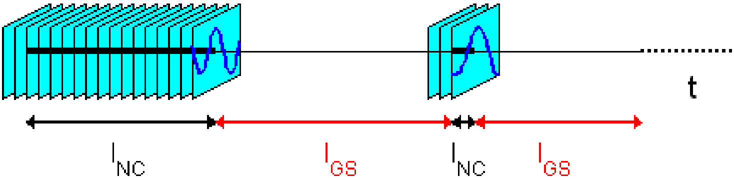
- Each L_2 projection requires all mesh points (expensive). Overcome when non linearity is algebraic
- Non–homogeneous BCs account for through a change of variable: requires some re–meshing when staggered grids are used
- Not suitable for transients
- Shows instability due to higher order mode truncation

Outline

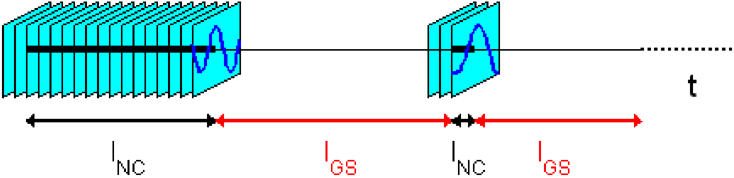
1. POD + Galerkin projection
   - POD + Galerkin projection (attractors)
   - POD + local Galerkin projection (transients+attractors)
2. Application to two model problems
   - The complex Ginzburg–Landau equation
   - The pulsating cavity
3. Conclusions and current work
   - Improving the method
   - Conclusions

POD + local Galerkin projection

- Rapún & Vega, J.Comput. Phys. 2010
- Developed to accelerate time dependent solvers in 1D
- Can be extended to treat steady flows (RANS solver) for varying parameter values
- Combines the NC solver with a Galerkin system in alternating time intervals I_{NC} and I_{GS}

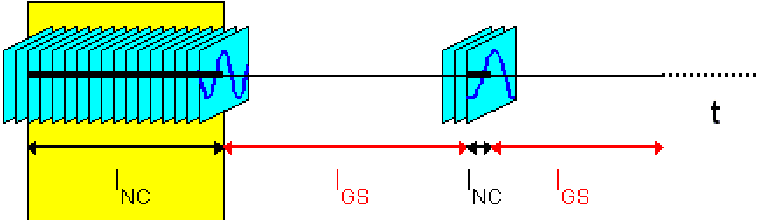


POD + local Galerkin projection



- Snapshots calculated in $0 < t < T$. Will the POD manifold also describe the dynamics for $t > T$?
  Yes, provided that some additional POD modes are retained (primary modes)
- Is it possible to estimate a priori the error of a Galerkin approximation?
  Yes, using some additional higher order modes (secondary modes)

POD + local Galerkin projection



System of semilinear parabolic equations

$$\mathcal{M}\partial_t \boldsymbol{q} = \mathcal{L}\boldsymbol{q} + \boldsymbol{f}(\boldsymbol{q}, t)$$

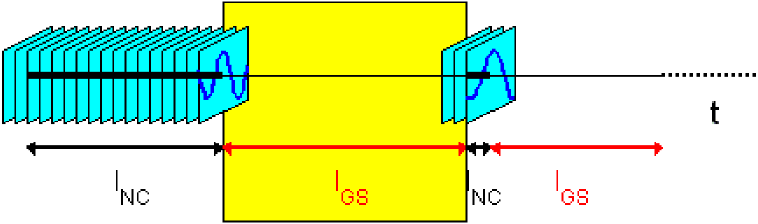
$\boldsymbol{q}$ is a state vector, $\mathcal{M}$ and $\mathcal{L}$ are linear, and $\boldsymbol{f}$ is nonlinear

Snapshots

We calculate N snapshots using the original NC

$$\boldsymbol{q}_1 = \boldsymbol{q}(\mathbf{x}, t_1), \ldots, \boldsymbol{q}_N = \boldsymbol{q}(\mathbf{x}, t_N)$$

POD + local Galerkin projection



POD

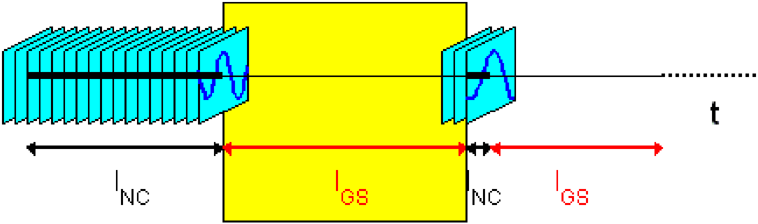
- Covariance matrix: $R_{ij} = \langle \boldsymbol{q}_i, \boldsymbol{q}_j \rangle$

$$\sum_{k=1}^{N} R_{ik} \alpha_j^k = (\sigma_j)^2 \alpha_j^i, \qquad j = 1, \ldots, N$$

- POD modes (orthonormal system)

$$\boldsymbol{Q}_j = \frac{1}{\sigma_j} \sum_{j=1}^{N} \alpha_j^k \boldsymbol{q}_k$$

POD + local Galerkin projection

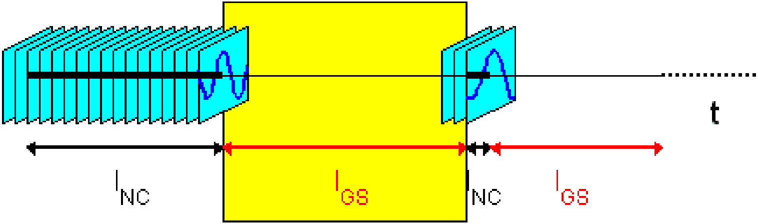


POD

- Snapshots in terms of POD modes $\boldsymbol{q}_\ell = \sum_{j=1}^{N} \sigma_j \bar{\alpha}_j^\ell \boldsymbol{Q}_j$
- For each $n < N$

$$\sum_{\ell=1}^{N} \|\boldsymbol{q}_\ell - \sum_{j=1}^{n} \sigma_j \bar{\alpha}_j^\ell \boldsymbol{Q}_j\|^2 = \sum_{j=n+1}^{N} (\sigma_j)^2$$

POD + local Galerkin projection



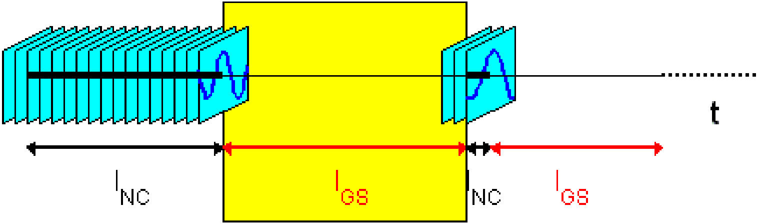
Projection onto the POD manifold

$$\boldsymbol{q} \approx \boldsymbol{q}_n = \sum_{j=1}^{n} a_j(t)\boldsymbol{Q}_j \quad \text{with } a_j = \langle \boldsymbol{Q}_j, \boldsymbol{q} \rangle$$

If $\|\boldsymbol{q} - \boldsymbol{q}_{n_1}\|$ is small for $n_1 > n$

$$E^n_{L_2} = \|\boldsymbol{q} - \boldsymbol{q}_n\| \approx \|\boldsymbol{q}_{n_1} - \boldsymbol{q}_n\| = \sqrt{\sum_{j=n+1}^{n_1} (a_j)^2} = E^{n_1}_n$$

POD + local Galerkin projection



Computation of $a_j(t)$

$$\mathcal{M}\partial_t \boldsymbol{q} = \mathcal{L}\boldsymbol{q} + \boldsymbol{f}(\boldsymbol{q}, t), \qquad \boldsymbol{q} \approx \boldsymbol{q}_n = \sum_{j=1}^{n} a_j(t)\boldsymbol{Q}_j$$

Projection onto the POD modes yields a GS:

$$\sum_{j=1}^{n} \langle \boldsymbol{Q}_i, \mathcal{M}\boldsymbol{Q}_j \rangle a_j'(t) = \sum_{j=1}^{n} \langle \boldsymbol{Q}_i, \mathcal{L}a_j(t) \rangle + \langle \boldsymbol{Q}_i, f(\sum_{k=1}^{n} a_k(t)\boldsymbol{Q}_k, t) \rangle$$

POD + local Galerkin projection

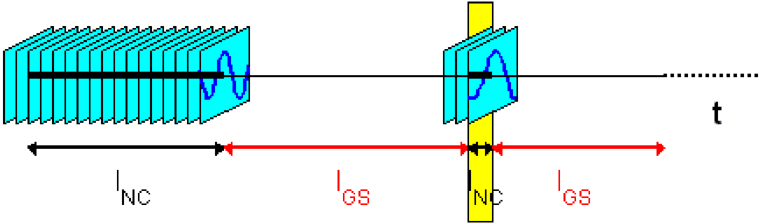
Our method:

1. Take $\nu = n, n_1, n_2$ s.t. $\sqrt{\sum_{j=\nu}^{N} \sigma_j{}^2} < \varepsilon, \varepsilon/100, \varepsilon/10000$
2. Solve the GS with n_1 and n_2 modes monitoring the errors

$$E_n^{n_1} = \sqrt{\sum_{j=n+1}^{n_1} a_j^2}, \qquad \widehat{E}_{n_1}^{n_2} = \|\boldsymbol{q}^n - \boldsymbol{q}^{n_2} - E_n^{n_2}\|$$

3. Stopping criteria
   - $E_n^{n_1} \geq \varepsilon$
   - $\widehat{E}_{n_1}^{n_2} \geq \varepsilon/100$ (consistency between the GSs in connection with higher order modes)
4. Two possibilities now:
   - If $\delta_{GS} < \delta_{GS,min}$: increase δ_{NC}, complete the NC solution and repeat (1-3)
   - Otherwise, continue

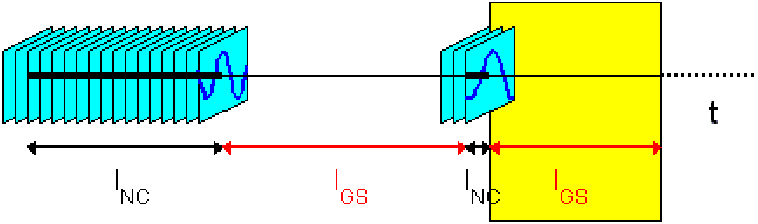
POD + local Galerkin projection



Computation of snapshots with the original NC

- Reconstruct $\boldsymbol{q}(t_{end}) = \sum_{j=1}^{n_2} a_j(t)\boldsymbol{Q}_j$ as initial condition
- Compute one (or a few) snapshots with the NC

POD + local Galerkin projection



Updating the POD manifold

POD to

$$\tilde{\nu}_1 \tilde{\boldsymbol{Q}}_1, \ldots, \tilde{\nu}_n \tilde{\boldsymbol{Q}}_n, \nu_1 \boldsymbol{Q}_1, \ldots, \nu_n \boldsymbol{Q}_n$$

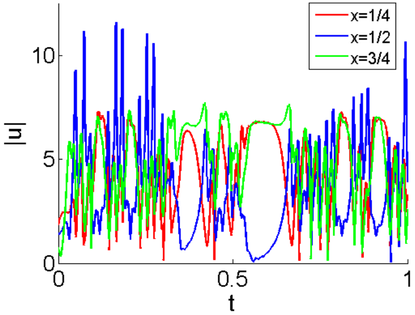
- $\tilde{\boldsymbol{Q}}_j$: POD modes in the last I_{GS}
- $\boldsymbol{Q}_j$: POD modes from the snapshots

Outline

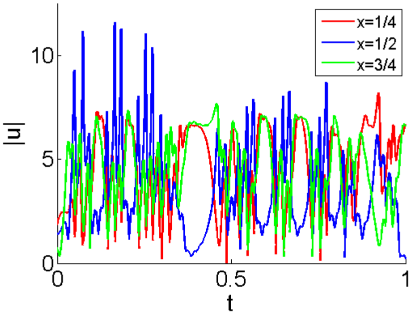
Complex Ginzburg–Landau equation

$$u_t = (1 + \imath\alpha)u_{xx} + \mu u - (1 + \imath\beta)|u|^2 u$$

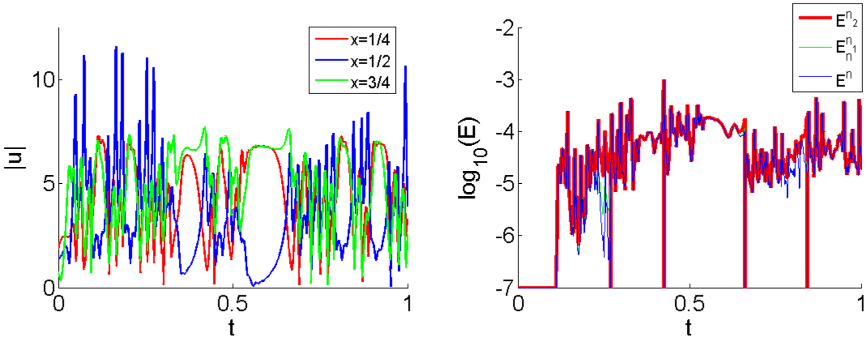
1000 mesh points



2000 mesh points



CGL in transient chaos with $(\alpha, \mu, \beta) = (-2, 90, 14)$

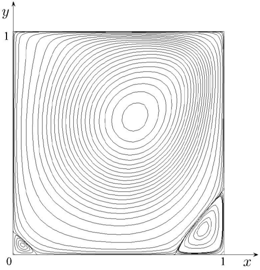


$$\frac{\text{DNS cost}}{\text{LPOD+GP cost}} = 8.54$$

Outline

Pulsating lid cavity

- Incompressible fluid dynamics in a 2D box whose upper wall is moving back and forth (2D NS eqn + non–homog BCs)
- A paradigm to test CFD
- Flow "complexity" at moderate Re



Governing equations

$$\nabla \cdot \boldsymbol{v} = 0$$

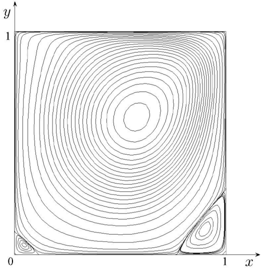
$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\nabla p + Re^{-1}\Delta v, \quad (x, y) \in]0, 1[^2$$

$$\boldsymbol{v}(0, y) = \boldsymbol{v}(1, y) = \boldsymbol{v}(x, 0) = \mathbf{0}, \quad \boldsymbol{v}(x, 1) = (h(t)g(x), 0)$$

$$g(x) = 16x^2(1 - x)^2, \quad h(x) = \sin(t); h(x) = \sin(\pi t/4)\cos(t/16)$$

Pulsating lid cavity

- Incompressible fluid dynamics in a 2D box whose upper wall is moving back and forth (2D NS eqn + non–homog BCs)
- A paradigm to test CFD
- Flow "complexity" at moderate Re

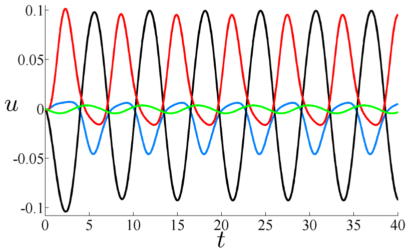


The pulsating cavity is quite demanding...

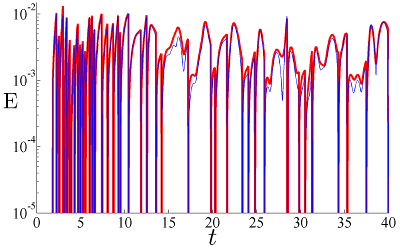
- Non–steadiness affects the boundary layer near the moving wall: increases the number of POD modes
- The bulk velocity is much smaller than the forcing velocity: requires stronger precision to maintain relative errors

Periodic motion on the upper wall, Re=100

Horizontal velocity

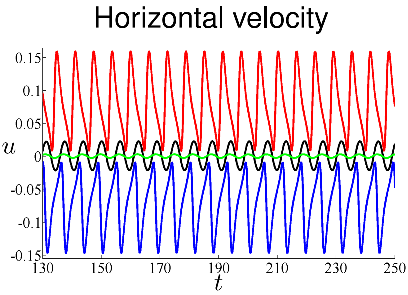


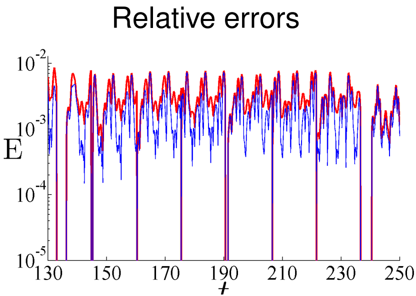
Relative errors



$$\frac{T}{\sum \delta_{NC}} = 9.3 \qquad \frac{\text{CPU time CFD}}{\text{CPU time LPOD+GP}} = 7.2$$

Periodic motion on the upper wall, Re=800

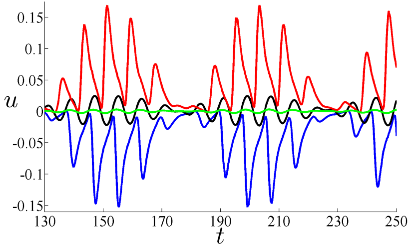




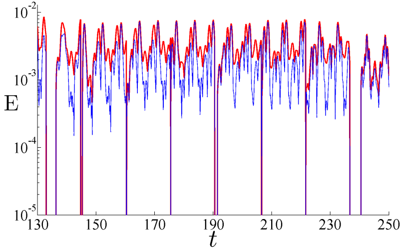
$$\frac{T}{\sum \delta_{NC}} = 7.2 \qquad \frac{\text{CPU time CFD}}{\text{CPU time LPOD+GP}} = 5.8$$

Quasi-periodic motion on the upper wall, Re=800

Horizontal velocity



Relative errors



$$\frac{T}{\sum \delta_{NC}} = 3.7 \qquad \frac{\text{CPU time CFD}}{\text{CPU time LPOD+GP}} = 2.6$$

Outline

1. POD + Galerkin projection
   - POD + Galerkin projection (attractors)
   - POD + local Galerkin projection (transients+attractors)
2. Application to two model problems
   - The complex Ginzburg–Landau equation
   - The pulsating cavity
3. Conclusions and current work
   - Improving the method
   - Conclusions

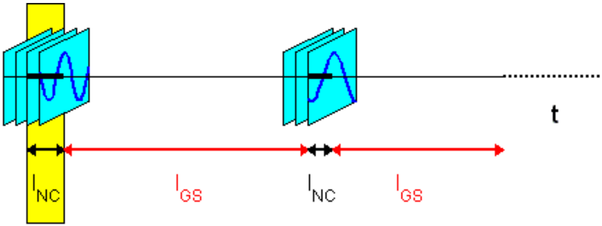
POD modes libraries

- The major computational effort in the LPOD+GP method is associated with the snapshots calculation in the first I_{NC} to construct the POD manifold
- The POD manifold depends only weakly on the particular values of the parameters of the problem

POD modes libraries

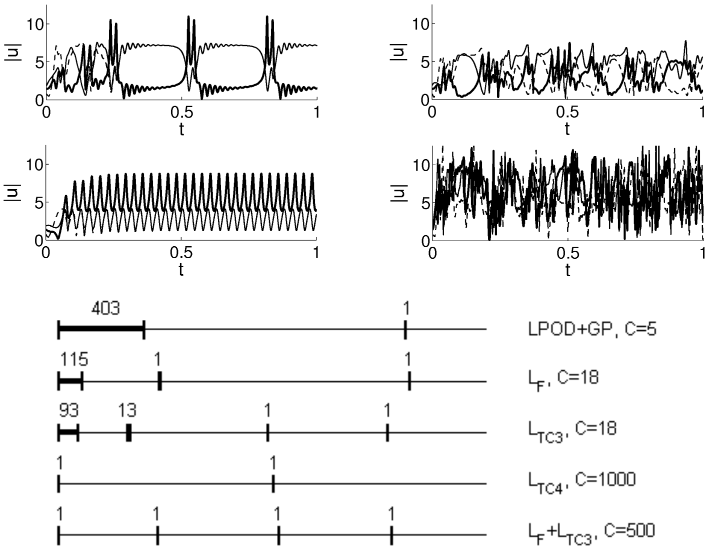
- Apply POD to a set of generic functions such as Fourier or orthogonal polynomials
- POD manifold resulting from other runs of the method for other parameter values
- Different libraries can be mixed up by applying POD to the joint sets of modes

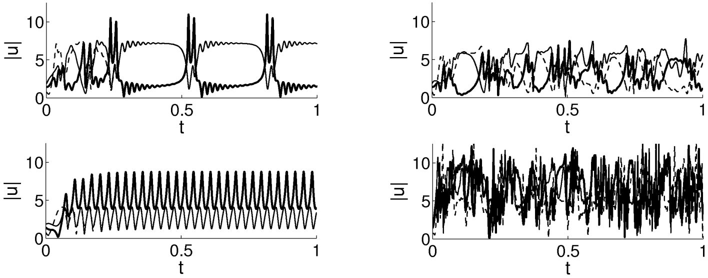
Modified method

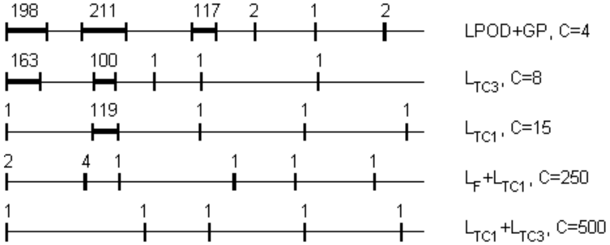


First I_{NC} interval

Use the POD modes library as in the basic method with the old POD manifold in the subsequent I_{NC} intervals


10
5
0
|u|
0
0.5
1
t
LPOD+GP, C=5
403
1
115
1
1
L_F, C=18
93
13
1
1
L_{TC3}, C=18
1
1
L_{TC4}, C=1000
1
1
1
1
L_F+L_{TC3}, C=500


|u|
10
5
0
0
0.5
1
t


198
211
117
2
1
2
LPOD+GP, C=4
163
100
1
1
1
L_{TC3}, C=8
1
119
1
1
1
L_{TC1}, C=15
2
4
1
1
1
1
L_F+L_{TC1}, C=250
1
1
1
1
1
$L_{TC1}+L_{TC3}$, C=500

Remarks

1. The selection of the POD modes libraries is not critical
2. Libraries resulting from more complex dynamics usually work better that those form simpler dynamics
3. The combination of POD modes libraries usually provides much better results
4. Our impression: many parabolic eqs might exhibit a POD manifold that approximately contains not only the attractors but also a significant part of the most relevant transient behaviors

Outline

Summarizing

- Efficient (cheap, robust, and precise) ROMs can be constructed using local POD+Galerkin projection
- Large theoretical/CPU compression, even using crude software (improvement in progress)
- Resulting errors comparable to CFD errors (checked using a spectral method)
- Both precise (projecting NS eqs) and rough (accounting for the CFD discretization) solvers can be dealt with
- Computational effort mainly due to first calculation of snapshots. Improved using snapshots libraries
- The selection of the POD modes libraries is not critical (a set of generic functions or from runs for different parameter values). Interesting for applications in industrial environments where solvers are run for a large amount of sets of parameter values